Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220147

Non-Admissibility in Abstract Argumentation

New Loop Semantics, Overview, Complexity Analysis

Wolfgang DVOŘÁK [a], Tjitze RIENSTRA [b], Leendert VAN DER TORRE [c,d], and Stefan WOLTRAN [a]

[a] *TU Wien*
[b] *Maastricht University*
[c] *University of Luxembourg*
[d] *Zhejiang University*

Abstract. In this paper, we give an overview of several recent proposals for non-admissible non-naive semantics for abstract argumentation frameworks. We highlight the similarities and differences between weak admissibility-based approaches and undecidedness-blocking approaches using examples and principles as well as a study of their computational complexity. We introduce a kind of strengthened undecidedness-blocking semantics combining some of the distinctive behaviours of weak admissibility-based semantics with the lower complexity of undecidedness-blocking approaches. We call it loop semantics, because in our new semantics, an argument can only be undecided if it is part of a loop of undecided arguments. Our paper shows how a principle-based approach and a complexity-based approach can be used in tandem to further develop the foundations of formal argumentation.

Keywords. abstract argumentation, semantics, complexity, weak admissibility

1. Introduction

Dung's admissibility-based (AB) semantics have been challenged in various ways, leading to a variety of new semantics [8,9,11,5]. These have been compared and classified on the basis of general principles as well as their computational complexity, such that the best semantics can be chosen for an application. Two desirable properties stand out. First, directionality together with SCC recursion lead to a kind of causal interpretation of attack [6] and allow for compositional computation [4]. Second, low computational complexity is not only advantageous for automated reasoning by artificial agents, but from the perspective of cognitive science, it also increases explainability by and for humans.

The best known non-admissible semantics are naive-based (NB) semantics. Under an NB semantics, each extension is a maximal conflict-free set of arguments. The most prominent example of an NB semantics is the CF2 semantics [3]. To illustrate the core idea, consider the framework (a) from Figure 1. In AB semantics, the only extension is the empty set, whereas under the CF2 semantics, b is accepted. To get the desirable properties of directionality and SCC-recursiveness, CF2 is defined in terms of a local

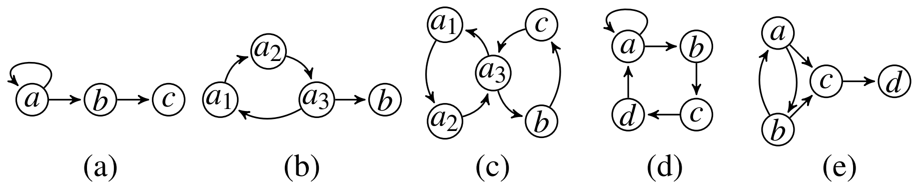

Figure 1. Five argumentation frameworks.

function that computes the maximal conflict-free subsets for each strongly connected component (or SCC) of a framework.

More recently, two new types of non-admissible semantics were introduced. They were motivated by the behaviour of AB semantics in examples such as framework (b) in Figure 1. Here, the set $\{b\}$ is not admissible since it does not defend itself from its attacker a_3. Nevertheless, one can argue that b is acceptable since a_3, being part of an odd-length cycle of arguments that are never accepted, does not pose an actual threat. Capturing this intuition thus requires a different notion of admissibility. The first type takes a reduction-based approach and is called weak admissibility (WA) [5]. The second type takes a labelling-based approach to define weaker acceptance criteria, called "undecidedness blocking" (UB) [11], which is analogous to "ambiguity blocking" and discussed in defeasible logics [15]. In contrast to the AB labelling approach, an undecided argument in UB may attack arguments that are labelled in. Further semantics that belong to the UB approach are the qualified and semi-qualified semantics [8]. These are defined by adapting the SCC recursive algorithm but keeping the base function admissible. The WA, UB and (semi-)qualified semantics all come in a grounded, complete and preferred flavour. These developments raise two questions: (1) How do these approaches compare in terms of examples, principles, and computational complexity? And (2) Which new kinds of semantics can be explored based on this overview?

Concerning the first question, the semantics of WA and UB approaches are remarkably similar for most benchmark examples. For example, they give the same results for the frameworks (b) and (c) from Figure 1. Moreover, both WA and UB preferred semantics are SCC-recursive and directional. The main distinction is in their computational complexity. As we show in this paper, the UB approach has a significantly lower computational complexity than the WA approach, for which we show PSPACE-completeness also for recently introduced variants (thus complementing the results in [14]).

The second question is how our analysis can be used to define new semantics. In particular, we are interested in approaches that combine the behaviour of WA semantics with the lower complexity of UB approaches. We define a new kind of semantics called loop semantics that extends UB with a new condition that ensures that arguments can only be undecided if they are part of a loop of undecided arguments. In this sense, the role of undecided arguments is to detect loops. We also define a notion of UB-admissibility, a concept that was missing thus far in the definition of UB semantics.

In this paper, we only consider non-admissible and non-naive based semantics. We do not consider the SCF2 semantics introduced by Cramer and van der Torre [7]. We focus our complexity analysis on complete and preferred variants of various semantics. Due to space limitations, we do not repeat all the variants of WA semantics described by Dauphin et al. [9], but we do include their semantics in the complexity analysis. For the same reason, for some of the proofs, we are only able to include proof sketches.

The layout of this paper is as follows. We first provide a brief overview of semantics based on weakly admissible and undecidedness blocking in Sections 2 and 3. We then present our new loop semantics in Section 4. All these semantics are illustrated using examples and principles. In Section 6 we provide a complexity overview of fifteen different kinds of non-admissible, non-naive semantics. We conclude in Section 7.

2. Weakly Admissible Semantics

An *argumentation framework* (abbreviated as AF) is a pair $F = (A, \rightarrow)$ where A is a set of arguments and $\rightarrow \subseteq A \times A$ the attack relation. We assume in this paper that A is finite. A set $E \subseteq A$ is *conflict-free* if there are no $x, y \in E$ such that $x \rightarrow y$. A set $E \subseteq A$ *defends* an argument $x \in A$ if for all $y \in A$ such that $y \rightarrow x$, there is a $z \in E$ such that $z \rightarrow y$ [12].

We focus in this paper on new variants of the *admissible*, *complete* and *preferred* semantics. The classical variants, denoted respectively by **ad**, **co** and **pr**, are defined as follows [12]. Let $F = (A, \rightarrow)$ be an AF. An **ad** extension of F is a set $E \subseteq A$ that is conflict-free and defends all its members. A **co** extension is an admissible extension that contains all arguments it defends. A **pr** extension is a maximal admissible extensions. In what follows we use, given a semantics σ, $\sigma(F)$ to denote the set of σ extensions of F.

Baumann, Brewka and Ulbricht [5] defined *weak admissibility* based on the principle that, given an AF $F = (A, \rightarrow)$ and set $E \subseteq A$, an argument needs to be defended by E only from arguments that are 'serious' in the sense that they appear in a weakly admissible set of the E*-reduct* of F. The E-reduct of F is denoted by F^E and defined by $F^E = (E^*, \rightarrow \cap (E^* \times E^*))$ where $E^* = A \setminus (E \cup E^+)$ and $E^+ = \{y \in A | \exists x \in E, x \rightarrow y\}$. Let $F = (A, \rightarrow)$ be an AF. A set $E \subseteq A$ is weakly admissible (i.e., $E \in \mathbf{ad}^w(F)$) if and only if E is conflict free and for every attacker y of E we have $y \notin \cup \mathbf{ad}^w(F^E)$. To define weakly complete and preferred semantics we first define *weak defence*.

Definition 1 *[5] Let $F = (A, \rightarrow)$ be an AF. A set $E \subseteq A$* weakly defends *a set $X \subseteq A$ whenever, for every attacker y of X, either E attacks y, or it is the case that $y \notin \cup \boldsymbol{ad}^w(F^E)$, $y \notin E$, and $X \subseteq X' \in \boldsymbol{ad}^w(F)$.*

Definition 2 *[5] Let $F = (A, \rightarrow)$ be an AF and $E \subseteq A$. The weakly complete and weakly preferred semantics $\boldsymbol{co}^w$ and $\boldsymbol{pr}^w$ are defined as follows: $E \in \boldsymbol{co}^w(F)$ iff $E \in \boldsymbol{ad}^w(F)$ and for every X such that $E \subseteq X$ that is w-defended by E, we have $X \subseteq E$; and $E \in \boldsymbol{pr}^w(F)$ iff E is $\subseteq$-maximal in $\boldsymbol{ad}^w(F)$.*

Dauphin et al. [9] defined several variants of the weak admissibility based semantics. We omit the definitions due to space constraints but we include the complete variants denoted $\mathbf{co}^{w\forall}$, $\mathbf{co}^{\exists}$ and $\mathbf{co}^{\forall}$ in Table 1 as well as our complexity analysis in Section 6.

We conclude this section by pointing out a notable difference between the weakly complete and preferred semantics with respect to the *directionality* principle [2]. Given an AF $F = (A, \rightarrow)$ and a set $U \subseteq A$, we use $F{\downarrow_U}$ to denote the AF $(U, \rightarrow \cap U \times U)$ and, given a set $X \subseteq 2^A$, we use $X{\downarrow_U}$ to denote the set $\{E \cap U | E \in X\}$. A semantics σ is directional if, for every AF $F = (A, \rightarrow)$ and every unattacked set U of F (i.e., any set $U \subseteq A$ such that $x \in U$ and $y \rightarrow x$ implies that $y \in U$) we have that $E \in \sigma(F){\downarrow_U} = \sigma(F{\downarrow_U})$. The preferred and complete semantics both satisfy directionality [2]. However, while the weakly preferred semantics also satisfies directionality [5], the weakly complete semantics does not [8].

Table 1. Various semantics applied to the AFs from Figure 1.

Semantics	(a)	(b)	(c)	(d)	(e)
co [12]	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$, $\{b,d\}$	$\emptyset$, $\{a,d\}$, $\{b,d\}$
pr [12]	$\emptyset$	$\emptyset$	$\emptyset$	$\{b,d\}$	$\{a,d\}$, $\{b,d\}$
$\mathbf{ad}^w$ [5]	$\emptyset, \{b\}$	$\emptyset$, $\{b\}$	$\emptyset$, $\{a_1\}$, $\{b\}$	$\emptyset$, $\{d\}$, $\{b\}$, $\{b,d\}$	$\emptyset$, $\{a\}$, $\{b\}$, $\{a,d\}$, $\{b,d\}$, $\{d\}$
$\mathbf{co}^w$ [5]	$\{b\}$	$\{b\}$	$\{a_1\}$, $\{b\}$	$\{b,d\}$	$\{a,d\}$, $\{b,d\}$, $\{d\}$
$\mathbf{pr}^w$ [5]	$\{b\}$	$\{b\}$	$\{a_1\}$, $\{b\}$	$\{b,d\}$	$\{a,d\}$, $\{b,d\}$
$\mathbf{co}^{w\forall}$ [9]	$\{b\}$	$\{b\}$	$\emptyset, \{a_1\}, \{b\}$	$\{b,d\}$	$\{a,d\}$, $\{b,d\}$, $\{d\}$
$\mathbf{co}^{\exists}$ [9]	$\{b\}$	$\{b\}$	$\{a_1\}, \{b\}$	$\{b,d\}$	$\{a,d\}$, $\{b,d\}$, $\{d\}$
$\mathbf{co}^{\forall}$ [9]	$\{b\}$	$\{b\}$	$\emptyset, \{a_1\}, \{b\}$	$\{b,d\}$	$\{a,d\}$, $\{b,d\}$, $\{d\}$
q-co	$\{b\}$	$\{b\}$	$\emptyset$	$\emptyset$, $\{b,d\}$	$\{a,d\}$, $\{b,d\}$, $\{c\}$
q-pr	$\{b\}$	$\{b\}$	$\emptyset$	$\{b,d\}$	$\{a,d\}$, $\{b,d\}$
sq-co	$\{b\}$	$\{b\}$	$\emptyset$	$\emptyset$, $\{b,d\}$	$\emptyset$, $\{a,d\}$, $\{b,d\}$
sq-pr	$\{b\}$	$\{b\}$	$\emptyset$	$\{b,d\}$	$\{a,d\}$, $\{b,d\}$
ub-co	$\emptyset, \{c\}, \{b\}$	$\emptyset$, $\{b\}$	$\emptyset$, $\{a_1\}$,$\{b\}$	$\emptyset$, $\{b,d\}$, $\{c\}$	$\emptyset$, $\{a,d\}$, $\{b,d\}$, $\{c\}$, $\{d\}$
ub-pr	$\{c\}$,$\{b\}$	$\{b\}$	$\{a_1\}$,$\{b\}$	$\{b,d\}$, $\{c\}$	$\{a,d\}$, $\{b,d\}$, $\{c\}$
ub2-co	$\{b\}$	$\{b\}$	$\emptyset$, $\{a_1\}$,$\{b\}$	$\emptyset$, $\{b,d\}$, $\{c\}$	$\{a,d\}$, $\{b,d\}$, $\{c\}$
ub2-pr	$\{b\}$	$\{b\}$	$\{a_1\}$,$\{b\}$	$\{b,d\}$, $\{c\}$	$\{a,d\}$, $\{b,d\}$
ub*-co	$\{b\}$	$\{b\}$	$\emptyset$, $\{a_1\}$,$\{b\}$	$\emptyset$, $\{b,d\}$	$\{a,d\}$, $\{b,d\}$, $\{c\}$
ub*-pr	$\{b\}$	$\{b\}$	$\{a_1\}$,$\{b\}$	$\{b,d\}$	$\{a,d\}$, $\{b,d\}$, $\{c\}$
ub2*-co	$\{b\}$	$\{b\}$	$\emptyset$, $\{a_1\}$,$\{b\}$	$\emptyset$, $\{b,d\}$	$\{a,d\}$, $\{b,d\}$, $\{c\}$
ub2*-pr	$\{b\}$	$\{b\}$	$\{a_1\}$,$\{b\}$	$\{b,d\}$	$\{a,d\}$, $\{b,d\}$

3. Existing Semantics Based On Undecidedness Blocking

We now review a number of recently proposed semantics that are based, like weak admissibility, on weaker acceptance criteria. They are defined in terms of labellings. A labelling L of an AF F is a function that maps each argument of F to a label $\mathtt{I}$ (in, or accepted), $\mathtt{O}$ (out, or rejected) or $\mathtt{U}$ (undecided). We use $\mathscr{L}(F)$ to denote the set of all labellings of F. A labelling-based semantics σ maps each AF F to a set $\mathscr{L}_\sigma(F) \subseteq \mathscr{L}(F)$. A labelling L corresponds to the extension containing all arguments labelled $\mathtt{I}$ by L.

The semantics we review in this section are based on an "undecidedness blocking" mechanism. While in an admissible labelling, an argument is labelled $\mathtt{I}$ only if all its attackers are labelled $\mathtt{O}$, these semantics allow $\mathtt{I}$-labelled arguments to be attacked by $\mathtt{U}$-labelled arguments. The semantics we discuss differ in the conditions under which this is allowed. As we will see, these semantics are, for most benchmark examples, remarkably similar to the weak admissibility based semantics from the previous section. We start with the *qualified* and *semi-qualified* schemes due to Dauphin et al. [8]. Both schemes rely on the decomposition of an AF into its SCCs (strongly connected components). We denote the set of the SCCs of F by $SCCS(F)$. Given an AF $F = (A, \rightarrow)$, an *outparent* of an SCC S of F is an argument $x \in A \setminus S$ such that $x \rightarrow y$ for some $y \in S$. We denote by $OP_F(S)$ the set of the outparents of S. Given a labelling $L \in \mathscr{L}(F)$, we denote by $L\downarrow_S$ the restriction of L to S and, given a set $X \subseteq \mathscr{L}(F)$, denote by $X\downarrow_S$ the set $\{L\downarrow_S \mid L \in X\}$.

Qualified Semantics The qualified scheme is based on the *SCC decomposability* principle, which states that the set of the labellings of an AF F is decomposable into the product of the labellings of each SCC S as a function of the labels of the outparents of S.

Definition 3 *An* AF with input *is a tuple* $(F, A_{in}, \rightarrow_{in}, L_{in})$ *where:* $F = (A, \rightarrow)$ *is an AF,* A_{in} *is a set of* input arguments *such that* $A \cap A_{in} = \emptyset$*,* $\rightarrow_{in} \subseteq A_{in} \times A$ *is an* input attack relation*, and* $L_{in} \in \mathscr{L}(A_{in})$ *is an* input labelling. *A* local function f *assigns to every AF with the input* $(F, A_{in}, \rightarrow_{in}, L_{in})$ *a set* $f(F, A_{in}, \rightarrow_{in}, L_{in}) \subseteq \mathscr{L}(F)$*. We say that* f represents *the semantics* σ *if for every AF* F*,* $L \in \mathscr{L}_\sigma(F)$ *if and only if* $\forall S \in SCCS(F)$*,* $L\downarrow_S \in f(F\downarrow_S, OP_F(S), \rightarrow \cap OP_F(S) \times S, L\downarrow_{OP_F(S)})$*. A semantics* σ *is* SCC decomposable *if it is represented by some local function.*

Examples of SCC decomposable semantics are the complete and preferred semantics. We denote by $f_{\mathbf{co}}$ and $f_{\mathbf{pr}}$ the local functions representing these semantics. Their definitions can be found in [1]. The qualified variant of an SCC decomposable semantics σ (denoted as **q-**σ) is based on applying the local function representing σ with one change: when determining the labellings of an SCC S, the label U for an outparent x of S is treated like the label O. Thus, if x is attacked by an U-labelled argument y, and x and y belong to different SCCs, then the undecidedness of x does not propagate to y.

Definition 4 *[8] Let* σ *be an SCC-decomposable semantics represented by the local function* f_σ*. We define the* qualified σ *(***q-**σ*) semantics as the semantics represented by the local function* $f_{\textbf{q-}\sigma}$ *defined by* $f_{\textbf{q-}\sigma}((A, \rightarrow), A_{in}, \rightarrow_{in}, L_{in}) = f_\sigma((A, \rightarrow), A_{in}, \rightarrow_{in}, L'_{in})$*, where* L'_{in} *is defined by* $L'_{in}(x) = \mathtt{I}$ *if* $L_{in}(x) = \mathtt{I}$*, and* $L'_{in}(x) = \mathtt{O}$ *if* $L_{in}(x) = \mathtt{O}$ *or* $L_{in}(x) = \mathtt{U}$*.*

Table 1 shows the **q-co** and **q-pr** extensions of the AFs from Figure 1. Here, we see that the **q-pr** extensions coincide with the weakly preferred extensions of AFs (a), (d) and (e), and that the **q-co** extensions coincide with the weakly complete extensions of AF (a). AF (e) demonstrates a crucial difference compared to weak admissibility. Here, $\{c\}$ is not weakly admissible because this set does not defend itself from a and b, both of which appear in weakly admissible sets of the $\{c\}$-reduct. However, under **q-co** semantics, the undecidedness of a and b does not propagate to c, as witnessed by the **q-co** labelling $(a = \mathtt{U}, b = \mathtt{U}, c = \mathtt{I}, d = \mathtt{O})$ with corresponding extension $\{c\}$.

Semi-qualified Semantics In the semi-qualified scheme, the label U of an outparent is treated like the label O, but *only if there is no other labelling in which that outparent is labelled* I. This is formalised using the notion of *weak SCC decomposability*.

Definition 5 *An* AF with total input *is a tuple* $(F, A_{in}, \rightarrow_{in}, L_{in}, S_{in})$ *where* $F, A_{in}, \rightarrow_{in}$ *and* L_{in} *are defined as in Definition 3,* $S_{in} \subseteq \mathscr{L}(A_{in})$*, and* $L_{in} \in S_{in}$*. We call* S_{in} *the* total input labellings *and* $L_{in} \in S_{in}$ *the* actual input labelling. *A* weak local function g *assigns to every AF with total input* $(F, A_{in}, \rightarrow_{in}, L_{in}, S_{in})$ *a set* $g(F, A_{in}, \rightarrow_{in}, L_{in}, S_{in}) \subseteq \mathscr{L}(F)$*. A weak local function* g *represents a semantics* σ *whenever, for every AF* F*,* $L \in \mathscr{L}_\sigma(F)$ *if and only if* $\forall S \in SCCS(F)$*,* $L\downarrow_S \in g(F\downarrow_S, OP_F(S), \rightarrow \cap OP_F(S) \times S, L\downarrow_{OP_F(S)}, \mathscr{L}_\sigma(F)\downarrow_{OP_F(S)})$*. A semantics* σ *is* weakly SCC-decomposable *if some weak local function represents* σ*.*

Definition 6 *[8] Let* σ *be an SCC-decomposable semantics. Let* f_σ *denote the local function that represents* σ*. We define the* semi-qualified σ *(or* **sq-**σ*) semantics as the semantics represented by the weak local function* $g_{\textbf{sq-}\sigma}$ *defined by* $g_{\textbf{sq-}\sigma}((A, \rightarrow), A_{in}, \rightarrow_{in}, L_{in}, S_{in}) = g_\sigma((A, \rightarrow), A_{in}, \rightarrow_{in}, L'_{in})$*, where* L'_{in} *is defined by:* $L'_{in}(x) = \mathtt{I}$ *if* $L_{in}(x) = \mathtt{I}$*;* $L'_{in}(x) = \mathtt{O}$ *if* $L_{in}(x) = \mathtt{O}$*;* $L'_{in}(x) = \mathtt{O}$ *if* $L_{in}(x) = \mathtt{U}$ *and there is no* $L \in S_{in}$ *such that* $L(x) = \mathtt{I}$*; and* $L'_{in}(x) = \mathtt{U}$ *if* $L_{in}(x) = \mathtt{U}$ *and there is some* $L \in S_{in}$ *such that* $L(x) = \mathtt{I}$*.*

Table 1 shows the **sq-co** and **sq-pr** extensions of the AFs shown in Figure 1. Note that the AF (e) no longer has a **sq-co** labelling where c is accepted. The **sq-co** and **sq-pr** semantics are different from the weak admissibility-based semantics in that the AF (c) only has an empty **sq-co** and **sq-pr** extension. This is because the (semi-)qualified scheme applies its base semantics unchanged to single-SCC AFs. The UB semantics that we present next addresses this problem. Before moving on we note that all variants of the (semi-)qualified semantics considered here satisfy directionality [8].

UB and UB2 Semantics Dondio and Longo [11] proposed a semantics based on the following definition. Note that this definition equals that of a standard complete labelling if we add as a third condition that an argument is labelled I only if all its attackers are labelled O. This semantics can thus be understood as a variant of the complete semantics where I-labelled arguments may also be attacked by U-labelled arguments. We refer to this semantics as *UB* semantics and define a complete and preferred variant.

Definition 7 *Let $F = (A, \rightarrow)$ be an AF. A* **ub-co** *labelling of F is a labelling L such that (1) $L(x) = \mathtt{O}$ iff for some $y \in A$ s.t. $y \rightarrow x$, $L(y) = \mathtt{I}$, and (2) if $L(x) = \mathtt{U}$ then for some $y \in A$ s.t. $y \rightarrow x$, $L(y) = \mathtt{U}$. A **ub-pr** labelling is a **ub-co** labelling that is maximal with respect to $\mathtt{I}$-labelled arguments.*

Looking at Table 1, we see that the weakly preferred and **ub-pr** extensions of the single-SCC AF (c) coincide. One way in which UB semantics diverges from weak admissibility is demonstrated by AF (a). Here, we have not only extension $\{b\}$ but also $\{c\}$ and $\emptyset$. This behaviour is due to the fact that, under UB semantics, the undecidedness of a may be blocked not only by b, but also by c or not at all. To avoid this behaviour, and to enforce the rule that undecidedness is blocked as early as possible, Dondio and Lungo propose to combine UB semantics with the SCC-recursive scheme [3]. These semantics, which we refer to as *UB2*, are directional by virtue of being SCC-recursive.

Definition 8 *Let $F = (A, \rightarrow)$ be an AF. The **ub2-co** semantics is defined by $L \in \mathscr{L}_{\textbf{ub2-co}}(F)$ iff the following conditions hold: If $|SCCS(F)| = 1$, then $L \in \mathscr{L}_{\textbf{ub-co}}(F)$; If $|SCCS(F)| > 1$, then for all $S \in SCCS(F)$: (1) $L{\downarrow}_{(S \setminus D_F(S,L))} \in \mathscr{L}_{\textbf{ub2-co}}(F{\downarrow}_{(S \setminus D_F(S,L))})$, and (2) $\forall x \in D_F(S,L), L(x) = \mathtt{O}$, where $D_F(S,L) = \{x \in S | \exists y \in A \setminus S, y \rightarrow x, L(y) = \mathtt{I}\}$ denotes the set of arguments in S that are attacked by an accepted argument not in S. The **ub2-pr** is defined similarly by replacing **co** with **pr** in this definition.*

Indeed, looking again at Table 1, the **ub2-pr** semantics coincides with the weakly preferred semantics in all of the examples we consider except for AF (d), where we see similar behaviour to that under **ub-pr** semantics: the undecidedness of a may be blocked not only by b, but also by c or not at all. The reason is that AF (d) consists of a single SCC, which leads to the same extensions under UB and UB2 semantics.

4. A New Semantics Based on Undecidedness Blocking

We now propose a new variant of UB semantics called *UB**. Our aim is to ensure that only arguments that are part of a loop can be undecided. While UB semantics requires that undecided arguments are *attacked by* an undecided argument, UB* semantics require that they also *attack* an undecided argument.

Definition 9 *Let $F = (A, \to)$ be an AF. A **ub*-co** labelling of F is a **ub-co** labelling of F such that, for all $x \in A$, if $L(x) = \mathtt{U}$, then $L(y) = \mathtt{U}$ for some $y \in A$ s.t. $x \to y$. A **ub*-pr** labelling is a **ub*-co** labelling that is maximal with respect to $\mathtt{I}$-labelled arguments.*

Looking again at Table 1, we see that **ub*-co** and **ub*-pr** semantics of the AF (d) no longer include $\{c\}$ as an extension. Unfortunately, this scheme is not yet sufficient to ensure that only arguments that are part of a loop can be undecided. It also allows arguments to be labelled $\mathtt{U}$ if they lie on a directed path between two cycles. For instance, the AF $(\{a,b,c\},\{(a,a),(a,b),(b,c),(c,c)\})$ has a **ub-co** labelling where b is labelled $\mathtt{U}$. Another problem is that the **ub*-pr** semantics is not directional. To see why, let F be the AF (e) from Figure 1. This AF has a **ub*-pr** labelling where a and b are $\mathtt{U}$, while $F{\downarrow}\{a,b\}$ does not have a **ub*-pr** labelling where a and b are $\mathtt{U}$. To ensure that an argument is undecided only if it is part of a cycle, we combine the UB* semantics with the SCC-recursive scheme. We refer to them as *loop semantics*. The complete and preferred loop semantics, denoted **ub2*-co** and **ub2*-pr**, are defined as in Definition 8 but replacing $\mathscr{L}_{\textbf{ub-co}}(F)$ and $\mathscr{L}_{\textbf{ub-pr}}(F)$ in condition 1 with $\mathscr{L}_{\textbf{ub*-co}}(F)$ and $\mathscr{L}_{\textbf{ub*-co}}(F)$. By virtue of being SCC-recursive, both of these semantics are directional. Furthermore, arguments are labelled $\mathtt{U}$ only if they are part of a loop. For example, the AF $(\{a,b,c\},\{(a,a),(a,b),(b,c),(c,c)\})$ does not posses a **ub*-co** labelling in which b is labelled $\mathtt{U}$.

5. UB-Admissibility

While Dondio et al. [11] define complete and preferred variants of the UB semantics, as well as a grounded variant (defined as an $\mathtt{U}$-maximal **ub-co** labelling) they do not define the concept UB admissibility. Note that the notion of “weak admissibility” defined in [10] is in fact a kind of UB completeness, and was renamed as such in [11]. Here we propose a notion of admissibility corresponding to the UB semantics. Having such a notion is useful because, as in Dung’s semantics, admissibility leads to local explanations as to why an argument is acceptable in AB semantics. That is, to check whether an argument belongs to a complete extension, we do not have to compute complete extension in their entirety: all we need to do is to find an admissible set containing the argument.

Standard admissibility is defined in terms of conflict-freeness and defence. In WA semantics, however, it is the other way around, in the sense that WA defence is defined in terms of weak admissibility. In both AB and WA approaches, complete extensions are defined in terms of (regular or weak) admissibility. In the definition of UB admissibility we introduce here, it is the other way around. We define UB admissibility in terms of complete UB extensions. The following definition defines UB-admissibility in terms of the UB-preferred semantics. Note that the same definition can also be applied in combination with the other semantics we have defined.

Definition 10 *Let $F = (A, \to)$ be an AF. An* UB-admissible *set of F is a set $E \subseteq A$ such that for some UB-preferred set E' of F we have that (1) $E \subseteq E'$; and (2) for all $x \in A$ such that $x \to E$ and $E' \to x$, we have $E \to x$.*

To illustrate, consider the AF (e) from Figure 1, which has the UB-admissible extensions $\emptyset$, $\{a\}$, $\{b\}$, $\{a,d\}$, $\{b,d\}$ and $\{c\}$, where only the latter three are UB-preferred. We leave a more detailed study of this notion of admissibility for future work.

Table 2. Complexity of weak-admissible based semantics compared to classical semantics ("c" is used as a shorthand for "complete")

σ	$Cred_\sigma$	$Scept_\sigma$	Ver_σ
co	NP-c	P-c	in P
pr	NP-c	Π_2^P-c	coNP-c
ad^w	PSPACE-c	trivial	PSPACE-c
co^w	PSPACE-c	PSPACE-c	PSPACE-c
pr^w	PSPACE-c	PSPACE-c	PSPACE-c
co$^{w\forall}$	PSPACE-c	PSPACE-c	PSPACE-c
co$^{\exists}$	PSPACE-c	PSPACE-c	PSPACE-c
co$^{\forall}$	PSPACE-c	PSPACE-c	PSPACE-c
q-co	NP-c	P-c	in P
q-pr	NP-c	Π_2^P-c	coNP-c

σ	$Cred_\sigma$	$Scept_\sigma$	Ver_σ
sq-co	Δ_2^P-c	in Δ_2^P	in Δ_2^P
sq-pr	Δ_2^P-c	Π_2^P-c	in Δ_2^P
ub-co	in P	in P	in P
ub-pr	in P	coNP-c	in P
ub2-co	NP-c	in P	in P
ub2-pr	NP-c	coNP-c	in P
ub*-co	NP-c	in P	in P
ub*-pr	NP-c	in Π_2^P	in coNP
ub2*-co	NP-c	in P	in P
ub2*-pr	NP-c	in Π_2^P	in coNP

6. Complexity Results

We start by briefly recalling some complexity classes. We assume that the reader is familiar with the basic concepts of computational complexity theory (see e.g. [13]) as well as the standard classes P, NP and coNP. In addition, we will consider the classes: $\Delta_2^P = P^{NP}$ of problems that can be solved in deterministic polynomial time when the algorithm has access to an NP oracle; $\Pi_2^P = coNP^{NP}$ of problems that can be solved in non-deterministic polynomial time when the algorithm has access to an NP oracle; and PSPACE of problems that can be solved using only the polynomial space of memory and exponential time. We have $P \subseteq NP/coNP \subseteq \Delta_2^P \subseteq \Pi_2^P \subseteq PSPACE$. The standard decision problems studied for an AF F and a semantics σ are: (1) Credulous/sceptical acceptance $Cred_\sigma/Scept_\sigma$ (does a given argument appear in at least one extension?); and (2) Verification Ver_σ: (does a given extension appear in $\sigma(F)$?) In what follows we study the complexity of these problems with regards to the semantics we consider (see Table 2).

Completeness notions based on BBU weak admissibility. To the best of our knowledge, the only existing complexity results for the semantics under consideration are those of [14] which show that the semantics of Baumann, Brewka and Ulbricht [5] are PSPACE-complete. By carefully inspecting the reductions in [14], we obtain that the different notions of weakly complete semantics of Dauphin et al. [9] are also PSPACE-hard, which is not surprising since these semantics are defined on top of weak admissibility. Moreover, it is easy to verify that the different notions of defence in the semantics of Dauphin et al. can be tested within PSPACE and thus PSPACE-completeness follows.

Qualified Semantics. We now turn to the complexity of qualified semantics, which are similar to the original semantics and thus it is not surprising that the complexity is unchanged. That is, we can verify a **q-co** labelling in polynomial time by processing the SCCs in topological order and, for each argument, check whether its label is valid w.r.t. its neighbours (differentiating between outparents and non-outparents). To verify a **q-pr** labelling, we also verify maximality, which can be done in coNP by the standard algorithm. The upper bounds for the reasoning problems are then arrived at by standard guessing and checking algorithms, taking into account that the unique minimal **q-co** can be computed via fixed-point iteration. The hardness results are obtained due to (a) the hardness results for *co* and *pr* semantics hold for strongly connected graphs and (b) on strongly connected graphs, qualified semantics coincide with the base semantics.[1]

Proposition 1 *The complexity results for* ***q-co*** *and* ***q-pr*** *semantics in Table 2 hold.*

Semi-qualified semantics. For semi-qualified semantics, verifying a labelling gets harder. In order to test whether a labelling is **sq-co**, it is not sufficient to validate the labels of the arguments with respect to the labels of the other arguments in the same labelling scheme, but we have to consider also the other labellings. However, once we know which of the arguments in the earlier SCCs are credulously accepted (and thus which are also labelled O at least once), we can verify the label of an argument in polynomial time.

That is, for all reasoning tasks, we first process the SCCs in a topological ordering, and for each SCC, we decide which of the arguments are credulously accepted. That is, for each argument a, we nondeterministically guess a labelling that labels a as I and labels all the arguments that are not in S or preceding SCCs as U. Such a labelling can then be verified in polynomial time, given that we already know which of the arguments in the preceding SCCs are credulously accepted. As we have to do this for each of the arguments, this part takes a linear number of NP oracle calls, i.e. we have a Δ_2^{P} algorithm for credulous acceptance. Given that we have computed all credulously accepted arguments, for **sq-co**, we can then run a polynomial time algorithm to verify a given labelling, while for **sq-pr**, we have to perform an additional maximality check which just requires an additional NP-oracle call. In total, this gives a Δ_2^{P} algorithm for the verification problem. The standard guessing and checking algorithm for sceptical acceptance now provides an Π_2^{P} algorithm for **sq-pr** while for **sq-co**, we can run a polynomial time fixed-point iteration to compute the unique minimal labelling, which results in a Δ_2^{P} algorithm.

The hardness of $Scept_{\textbf{sq-pr}}$ holds because the hardness results for *pr* semantics hold even for single-SCC AFs. Next, consider the Δ_2^{P}-hardness of $Cred_{\textbf{sq-pr}} = Cred_{\textbf{sq-co}}$. This is by a reduction from the Δ_2^{P}-complete problem of deciding whether for a propositional formula in CNF φ given by a set of clauses C over atoms $x_1,\dots,x_n$, the lexicographical maximum satisfying assignment of ϕ sets x_n to true. The reduction builds n SCCs, each corresponding to an adaptation of the standard translation from SAT to AFs, but which are in a linear order. The AF $G_\varphi = (A,R)$ is constructed as follows: $A = \{x_i^j \mid 1 \leq i \leq j \leq n\} \cup \{\bar{x}_i^j \mid 1 \leq i,j \leq n, i \neq j\} \cup \{c^j \mid 1 \leq j \leq n, c \in C\} \cup \{t^j, b^j \mid 1 \leq j \leq n\}$; and

$$\begin{aligned}
R =&\{(x_i^j,\bar{x}_i^j),(\bar{x}_i^j,x_i^j) \mid x_i^j \in A\} \cup \{(x_i^j,c^j) \mid x_i^j \in A, x_i \in c \in C\} \cup \\
&\{(\bar{x}_i^j,c^j) \mid \bar{x}_i^j \in A, \neg x_i \in c \in C\} \cup \{(c^j,t^j),(t^j,b^j),(b^j,b^j) \mid 1 \leq j \leq n\} \cup \\
&\{(b^j,x_i^j) \mid x_i^j \in A\} \cup \{(b^j,\bar{x}_i^j) \mid \bar{x}_i^j \in A\} \cup \\
&\{(x_i^i,c^j) \mid x_i \in c \in C, 1 \leq i \leq j \leq n)\} \cup \{(x_i^i,\bar{x}_i^j) \mid 1 \leq i < j \leq n\}.
\end{aligned}$$

Notice that the upper index of the arguments denote the SCC they belong to, and only the last line in the definition of the attack relation introduces attacks between SCCs.

[1]Note that these hardness proofs construct AFs with an empty grounded extension. Then, one can add a self-attacking argument g that symmetrically attacks all other arguments, without changing the complete extensions.

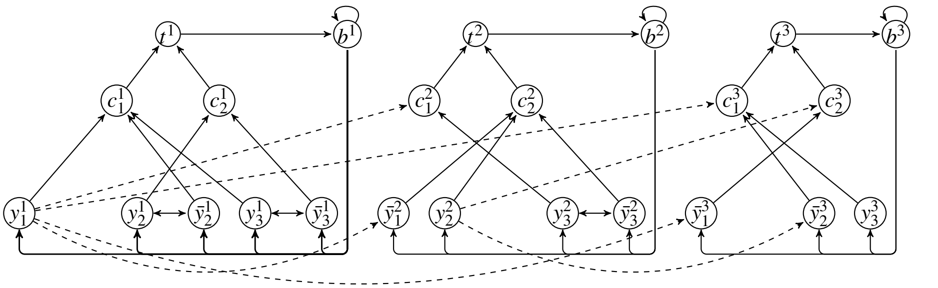

Figure 2. Illustration of the reduction G_φ for the formula φ with clauses $\{\{y_1, \neg y_2, y_3\}, \{\neg y_1, y_2, \neg y_3)\}$. Attacks between different SCCs are highlighted as dashed lines

The intuition is as follows. In the first SCC, we test whether some assignment sets x_1 to true. If so, we fix this assignment by adding x_1^1 to the extension, which then attacks all arguments $\bar{x}_1^j$. If not, all arguments in the first SCC are labelled U, and we have to pick $\bar{x}_1^j$ (we may assume that C contains clauses $(x_i, \neg x_i)$ in the latter SCCs), we proceed like that, and in the j-th SCC, we check whether some assignment sets x_j to true given the already fixed assignment on earlier variables. One can show that x_n is true in the lexicographic-maximum-satisfying assignment of φ iff x_n^n is credulously accepted in G_φ.

Proposition 2 *The complexity results for* ***sq-co*** *and* ***sq-pr*** *semantics in Table 2 hold.*

UB Semantics. While **ub-**semantics requires that defended arguments are labelled I, they do not require that all I-labelled arguments are defended. It turns out that this lack of admissibility reduces the complexity significantly. First, notice that by definition in UB-complete labellings: (a) if an argument a is labelled I, then all arguments attacked by a must be labelled O; and (b) if all attackers of an argument a are labelled O, then a must be labelled I. We can compute UB-complete labellings by starting with a set S of I-labelled arguments, and then use the two rules from above to propagate labels until either we obtain that an argument must be both labelled I and O or a fixed point is reached. In the former cases, we have that there is no UB-complete labelling that labels all the arguments in S as I. In the latter case, we label the remaining arguments U to obtain UB-complete labelling. By that, we have that the grounded labelling is the unique minimal **ub-co** labelling, and thus sceptical acceptance is P-complete. To decide on credulous acceptance w.r.t. **ub-co** (and **ub-**pr), one can fix the label of the query argument as I and apply the characteristic function until a fixed point is reached. If the result is conflict free, the argument is credulously accepted, otherwise it is not accepted. The conditions for a **ub-co** labelling can be easily checked in polynomial time, When verifying **ub-pr** labellings, we have to also check the maximality condition. Let S be the set of I-labelled arguments. We can now test for each U-labelled argument a whether $S \cup \{a\}$ is contained in some UB-complete labelling. This can be done simply by the above-mentioned fixed-point iteration and is thus in polynomial time. Given that verification is in P, we can solve $Scept_{\textbf{ub-pr}}$ with the standard guessing and checking approach in coNP. We next show that $Scept_{\textbf{ub-pr}}$ is also coNP-hard. To this end, consider the following adaptation of standard reduction (cf. Figure 2). Given a propositional formula φ in CNF given by a set of clauses C over atoms Y, we define φ as $F_\varphi = (A, R)$ (cf. Figure 2), where

$A = \{\varphi, \bar{\varphi}_1, \bar{\varphi}_2\} \cup C \cup Y \cup \bar{Y}$ and $R = \{(c, \bar{\varphi}_1) \mid c \in C\} \cup \{(l,c),(c,c) \mid l \in c, c \in C\} \cup \{(x,\bar{x}),(\bar{x},x) \mid x \in Y\} \cup \{(\bar{\varphi}_1, \varphi),(\varphi, \bar{\varphi}_2),(\bar{\varphi}_2, \bar{\varphi}_1)\}$. If all clause arguments c_i are labelled `O`, none of the arguments in the cycle can be accepted. Otherwise, if at least one c_i remains `U`, we can accept φ set $\bar{\varphi}_2$ as `O` and $\bar{\varphi}_1$ as `U`. We thus have that the argument φ is sceptically accepted iff formula φ is unsatisfiable.

Proposition 3 *The complexity results for* ***ub-co*** *and* ***ub-pr*** *semantics in Table 2 hold.*

UB2 semantics. We now turn to the SCC-recursive variants of the **ub-**semantics, **ub2-**semantics. In order to verify a labelling, we can follow the SCC-recursive schema and apply the verification of the base semantics in the base case. Since verification of **ub-**semantics is in polynomial time, we obtain that **ub2-**semantics can also be verified in polynomial time. The NP/coNP-membership of credulous/sceptical reasoning is then verified by the standard guessing and checking algorithms, and the matching hardness results are verified by standard reductions for credulous and sceptical acceptance.

Proposition 4 *The complexity results for* ***ub2-co*** *and* ***ub2-pr*** *semantics in Table 2 hold.*

UB and UB*2 semantics.* Similarly, for the UB-complete semantics, we can verify UB*-complete and UB*2-complete extensions in polynomial time, and thus the remaining upper bounds are obtained by standard procedures.

Now, consider the verification of an UB*-preferred extension. The additional condition that an `U`-labelled argument has to attack an `U`-labelled argument allows for a similar behaviour to that of standard Dung complete and preferred semantics. Consider an argument a that is labelled `I` and an argument b that attacks only argument a. We have that b cannot be labelled `U` or `I` and thus has to be labelled `O`. That is, we have to find an argument c that can be labelled `I` and defends a against b. With this observation, one can easily adapt the standard translations such that the NP/coNP hardness results for the reasoning tasks of complete and preferred semantics also transfer to UB* complete and UB* preferred semantics (even for strongly connected graphs).

Proposition 5 *The complexity results for* ***ub*-co****,* ***ub*-pr ub2*-co*** *and* ***ub2*-pr*** *semantics in Table 2 hold.*

7. Conclusion

We reviewed several recent proposals for non-admissible non-naive semantics for abstract argumentation and studied their complexity. We focused in particular on semantics that behave similar to the weakly complete and preferred semantics, but are based on undecidedness blocking mechanisms. Our complexity results (Table 2) show that this approach has significantly lower complexity than the weak admissibility based approach. We also defined a variant called *loop semantics*, that (1) assigns the label `U` only to arguments that are part of a loop, and (2) allows arguments labelled `I` to be attacked by `U`-labelled arguments. Unlike the weakly complete semantics, the complete variant of this semantics satisfies directionality. We plan to investigate the properties of this new semantics, as well as the derived notion of UB-admissibility, in future work.

Acknowledgements

This work was funded by the Austrian Science Fund (FWF) under grant Y698 and by the Vienna Science and Technology Fund (WWTF) under grant ICT19-065. Leendert van der Torre acknowledges financial support from the Fonds National de la Recherche Luxembourg (INTER/Mobility/19/13995684/DLAl/van der Torre).

References

[1] Pietro Baroni, Guido Boella, Federico Cerutti, Massimiliano Giacomin, Leendert van der Torre, and Serena Villata. On the input/output behavior of argumentation frameworks. *Artificial Intelligence*, 217:144–197, 2014.

[2] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *Knowledge Engineering Review*, 26(4):365–410, 2011.

[3] Pietro Baroni, Massimiliano Giacomin, and Giovanni Guida. SCC-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence*, 168(1-2):162–210, 2005.

[4] Pietro Baroni, Massimiliano Giacomin, and Beishui Liao. Locality and modularity in abstract argumentation. In *Handbook of formal argumentation*, pages 937–979. London, UK: College Publications, 2018.

[5] Ringo Baumann, Gerhard Brewka, and Markus Ulbricht. Shedding new light on the foundations of abstract argumentation: Modularization and weak admissibility. *Artificial Intelligence*, 310:103742, 2022.

[6] Alexander Bochman. *Explanatory Nonmonotonic Reasoning*, volume 4 of *Advances in Logic*. World Scientific, 2005.

[7] Marcos Cramer and Leendert van der Torre. SCF2 - an argumentation semantics for rational human judgments on argument acceptability. In *Proceedings of the 8th Workshop on Dynamics of Knowledge and Belief (DKB-2019) and the 7th Workshop KI & Kognition (KIK-2019)*, pages 24–35, 2019.

[8] Jeremie Dauphin, Tjitze Rienstra, and Leendert van der Torre. A principle-based analysis of weakly admissible semantics. In *Proceedings of COMMA2020*, pages 167–178. IOS Press, 2020.

[9] Jérémie Dauphin, Tjitze Rienstra, and Leendert van der Torre. New weak admissibility semantics for abstract argumentation. In *Logic and Argumentation - Proceedings of CLAR2021*, volume 13040 of *Lecture Notes in Computer Science*, pages 112–126. Springer, 2021.

[10] Pierpaolo Dondio. Weakly-admissible semantics and the propagation of ambiguity in abstract argumentation semantics. Technical report, Technological University Dublin, 2019.

[11] Pierpaolo Dondio and Luca Longo. Weakly complete semantics based on undecidedness blocking. *arXiv preprint arXiv:2103.10701*, 2021.

[12] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357, 1995.

[13] Wolfgang Dvořák and Paul E. Dunne. Computational problems in formal argumentation and their complexity. In *Handbook of Formal Argumentation*, pages 631–687. London, UK: College Publications, 2018.

[14] Wolfgang Dvořák, Markus Ulbricht, and Stefan Woltran. Recursion in abstract argumentation is hard - on the complexity of semantics based on weak admissibility. In *Proceedings of AAAI2021*, pages 6288–6295, 2021.

[15] Guido Governatori, Michael J. Maher, Grigoris Antoniou, and David Billington. Argumentation semantics for defeasible logic. *Journal of Logic and Computation*, 14(5):675–702, 2004.